



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/750,297	12/31/2003	Richard M. Shupak	MSFT-2568/307781.01	1690
41505 7590 11/26/2007 WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION) CIRA CENTRE, 12TH FLOOR 2929 ARCH STREET PHILADELPHIA, PA 19104-2891			EXAMINER SCHMIDT, KARI L	
			ART UNIT 2139	PAPER NUMBER
			MAIL DATE 11/26/2007	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

## Office Action Summary

Application No.

10/750,297

Applicant(s)

SHUPAK ET AL.

Examiner

Kari L. Schmidt

Art Unit

2139

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 26 October 2007.
- 2a) ☐ This action is FINAL. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1,2,5,6,10,11,14,15,19,20,23 and 27-36 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☐ Claim(s) 1,2,5,6,10,11,14,15,19,20,23 and 27-36 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 31 December 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- ☒ Notice of References Cited (PTO-892)
- ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- ☒ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_
- ☐ Notice of Informal Patent Application
- ☐ Other: \_\_\_\_\_

## **DETAILED ACTION**

### ***Continued Examination Under 37 CFR 1.114***

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 9/4/2007 has been entered.

### ***Notice to Applicant***

This communication is in response to the amendment filed on 10/26/2007. Claims 1,2,5,6,10,11,14,15,19,20,23 and 27-36 remain pending. Claims 3-4, 7-9, 12-13, 16-18, 21-22, and 24-26 have been canceled.

### ***Response to Arguments***

With regard to Applicant's arguments, it is respectfully submitted that the Examiner has applied new prior art to the amended features of claims (1, 5, 6, 10, 14, 15,19, and 23). Applicant's arguments have been considered but are moot in view of the new ground(s) of rejection.

***Claim Rejections - 35 USC § 102***

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1,2,5,6,10,11,14,15,19,20,23 and 27-36 are rejected under 35

U.S.C. 102(b) as being anticipated by Lueh (US 6,658,657 B1).

Claims 1, 10, 19

Lueh discloses method, system and medium of processing runtime functions, comprising: receiving a call to a runtime function; determining associated data from the call to the runtime function; determining a target address from the associated data; comparing the target address with a reference list of valid target addresses; and if the target address is found on the reference list of valid target addresses then executing the target (see at least, column 2, lines 9-45, column 4, lines 55-65: the virtual method gets inlined, the compiler such as JIT compiler generates a run-time test to verify if the inlined callee is the right instance to be invoked. The run-time test is typically implemented by checking the vtable or by checking foo of the actual target address of the method invocation. Checking the vtable involves comparing the object's vtable with the vtable of the class of the inlined method. If the comparison is successful (i.e. object matches the vtable of the class of the inlined method) it is safe to execute the inlined code because the inlined method will be dynamically dispatched at runtime. If the

comparison fails (i.e. object does not match the vtable of the class of the inlined method) the conventional dispatching code sequence is executed to invoke the virtual method call...).

Claims 2, 11, 20

Lueh discloses the method of claim 1, wherein the step of determining the associated data comprises accessing data in a data structure connected with the runtime function and calculating the associated data based on the accessed data (see at least, column 1, lines 12-65, Figures 1 and 2, column 4, lines 45--65).

Claims 5, 14, 23

Lueh discloses the method of claim 1 comprising the step of generating the reference list of valid target addresses at compiler and link time (see at least, column 2, lines 9-45, column 4, lines 55-65: the virtual method gets inlined, the compiler such as JIT compiler generates a run-time test to verify if the inlined callee is the right instance to be invoked. The run-time test is typically implemented by checking the vtable or by checking foo of the actual target address of the method invocation. Checking the vtable (reference list) involves comparing the object's vtable with the vtable of the class of the inlined method. If the comparison is successful (i.e. object matches the vtable of the class of the inlined method) it is safe to execute the inlined code because the inlined method will be dynamically dispatched at runtime. If the comparison fails (i.e. object does not match the vtable of the class of the inlined method) the conventional dispatching code

sequence is executed to invoke the virtual method call...).

Claims 6, 15, 27

Lueh discloses the method of claim 1 comprising the step of generating the reference list of valid target addresses at runtime (see at least, column 2, lines 9-45, column 4, lines 55-65: the virtual method gets inlined, the compiler such as JIT compiler generates a run-time test to verify if the inlined callee is the right instance to be invoked. The run-time test is typically implemented by checking the vtable or by checking foo of the actual target address of the method invocation. Checking the vtable (reference list) involves comparing the object's vtable with the vtable of the class of the inlined method. If the comparison is successful (i.e. object matches the vtable of the class of the inlined method) it is safe to execute the inlined code because the inlined method will be dynamically dispatched at runtime. If the comparison fails (i.e. object does not match the vtable of the class of the inlined method) the conventional dispatching code sequence is executed to invoke the virtual method call...).

Claims 28, 31, 34

Lueh discloses the method of claim 1 comprising the step of storing the target address in a caller provided location (see at least, column 2, lines 9-45, column 4, lines 55-65: the virtual method gets inlined, the compiler such as JIT compiler generates a run-time test to verify if the inlined callee is the right instance to be invoked. The run-time test is typically implemented by checking the vtable or by checking foo of the actual target

address of the method invocation. Checking the vtable involves comparing the object's vtable with the vtable of the class of the inlined method. If the comparison is successful (i.e. object matches the vtable of the class of the inlined method) it is safe to execute the inlined code because the inlined method will be dynamically dispatched at runtime. If the comparison fails (i.e. object does not match the vtable of the class of the inlined method) the conventional dispatching code sequence is executed to invoke the virtual method call... vtable is stored in memory which is the address list and it is stored in memory at run-time).

Claims 29, 32, 35

Lueh discloses the method of claim 1 comprising the step of terminating execution if the target address is not found on the reference list of valid target addresses (see at least, column 2, lines 9-45, column 4, lines 55-65: the virtual method gets inlined, the compiler such as JIT compiler generates a run-time test to verify if the inlined callee is the right instance to be invoked. The run-time test is typically implemented by checking the vtable or by checking foo of the actual target address of the method invocation.

Checking the vtable involves comparing the object's vtable with the vtable of the class of the inlined method. If the comparison is successful (i.e. object matches the vtable of the class of the inlined method) it is safe to execute the inlined code because the inlined method will be dynamically dispatched at runtime. If the comparison fails (i.e. object does not match the vtable of the class of the inlined method) the conventional dispatching code sequence is executed to invoke the virtual method call...; column 7,

lines 35-54).

Claims 30, 33, 36

Lueh disclose the method of claim 1 comprising the step of storing the reference list of valid target addresses in memory (see at least, column 2, lines 9-45, column 4, lines 55-65: the virtual method gets inlined, the compiler such as JIT compiler generates a run-time test to verify if the inlined callee is the right instance to be invoked. The run-time test is typically implemented by checking the vtable or by checking foo of the actual target address of the method invocation. Checking the vtable involves comparing the object's vtable with the vtable of the class of the inlined method. If the comparison is successful (i.e. object matches the vtable of the class of the inlined method) it is safe to execute the inlined code because the inlined method will be dynamically dispatched at runtime. If the comparison fails (i.e. object does not match the vtable of the class of the inlined method) the conventional dispatching code sequence is executed to invoke the virtual method call...).

***Conclusion***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kari L. Schmidt whose telephone number is 571-270-1385. The examiner can normally be reached on Monday - Friday: 7:30am - 5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ayaz Sheikh can be reached on 571-272-3795. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

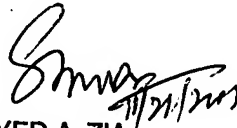


Application/Control Number:  
10/750,297  
Art Unit: 2139

Page 8

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

KS

  
SYED A. ZIA  
PRIMARY EXAMINER